Accurate discretization of diffusion in the LS-STAG cut-cell method using diamond cell techniques

B. Portelenelle, Olivier Botella, Y. Cheny

To cite this version:

B. Portelenelle, Olivier Botella, Y. Cheny. Accurate discretization of diffusion in the LS-STAG cut-cell method using diamond cell techniques. Computers and Fluids, 2019, 189, pp.34-45. 10.1016/j.compfluid.2019.05.005 . hal-02173850

HAL Id: hal-02173850
https://hal.univ-lorraine.fr/hal-02173850v1

Submitted on 25 Oct 2021

Accurate discretization of diffusion in the LS-STAG cut-cell method using diamond cell techniques

B. Portelenelle[a], O. Botella[a,*], Y. Cheny[a]

[a]*LEMTA - Université de Lorraine- 2 avenue de la Forêt de Haye, TSA 60604 - 54518 Vandœuvre CEDEX, France*

Abstract

In cut-cell methods for fluid flow in complex geometries such as the LS-STAG method (Nikfarjam *et al.* Comput Phys Commun 226:67-80, 2018), mesh non-orthogonality near the immersed boundary makes inaccurate 2-point formulas for computing the face-normal gradients in diffusive fluxes. A way to improve the accuracy is to compute the whole solution gradient at the cut-cell faces, thus decomposing the flux as an orthogonal contribution (using a standard 2-point formula) and non-orthogonal correction (using data at cell vertices). The solution at cut-cell vertices is then interpolated from cell-centered data and boundary conditions. This gradient reconstruction technique is commonly denominated "diamond cell method".

The diamond cell method is implemented in the LS-STAG code for cell-centered data (temperature equation in 2D geometries) and face-centered data (Navier-Stokes equations in 3D-extruded geometries) using various interpolation schemes for vertex reconstruction. The accuracy of the diamond cell discretization is firmly assessed on a series of 2D benchmark problems (Taylor-Couette solution, natural convection from a cylinder in an enclosure) by inspecting the formal order of accuracy and the heat flux distribution at the immersed boundary. Finally, the diamond cell technique is employed for enhancing the accuracy of the velocity gradients in 3D-extruded geometries which were previously computed with 2-point formulas. These computations illustrate the versatility of the diamond cell technique, which can be applied to other cut-cell methods.

Keywords: Computational fluid dynamics (CFD), Incompressible Flows, Heat Transfer, Cut-Cell Method, Finite Volume Method, Gradient Discretisation.

---

1. Introduction

The LS-STAG method [1] is a Cartesian method for incompressible flow computations in irregular geometries which aims at discretizing accurately the flow equations in the cut-cells, *i.e.* cells of complex polyhedral shape formed by the intersection of the Cartesian mesh with the immersed boundary. In the three basic types of 2D cut-cells (pentagonal, trapezoidal and triangular, see Fig. 1(top)), the discretization is designed such as the global invariants of the flow (total mass, momentum and kinetic energy) are preserved at the discrete level. The LS-STAG discretization in the cut-cells is consistent with the MAC discretization used in Cartesian fluid cells, and has the ability to preserve the 5-point Cartesian structure of the stencil, resulting in a highly

---

[*] Corresponding author

Email addresses: `brice.portelenelle@univ-lorraine.fr` (B. Portelenelle), `olivier.botella@univ-lorraine.fr` (O. Botella), `yoann.cheny@univ-lorraine.fr` (Y. Cheny)

Preprint submitted to Journal of LaTeX Templates *April 30, 2019*

computationally efficient method. The LS-STAG method has successfully been applied to Newtonian flows at moderate Reynolds number in fixed and moving geometries [1], pseudoplastic flows [2], and viscoelastic flows [3]. In a recent paper [4], the LS-STAG discretization has been extended to 3D configurations with translational symmetry in the z direction (subsequently called “3D-extruded” configurations), where only the 4 types of cells depicted in Fig. 1(bottom) are present. In such geometries, only extruded 2D cut-cells are present, which enables us to easily extend the principles of the energy-conserving discretization of Ref. [1] to the cut-cells of Fig. 1(bottom) : the discretization of the continuity equation, pressure gradient, convective fluxes, normal stresses are straightforward extensions of the 2D LS-STAG formulas.

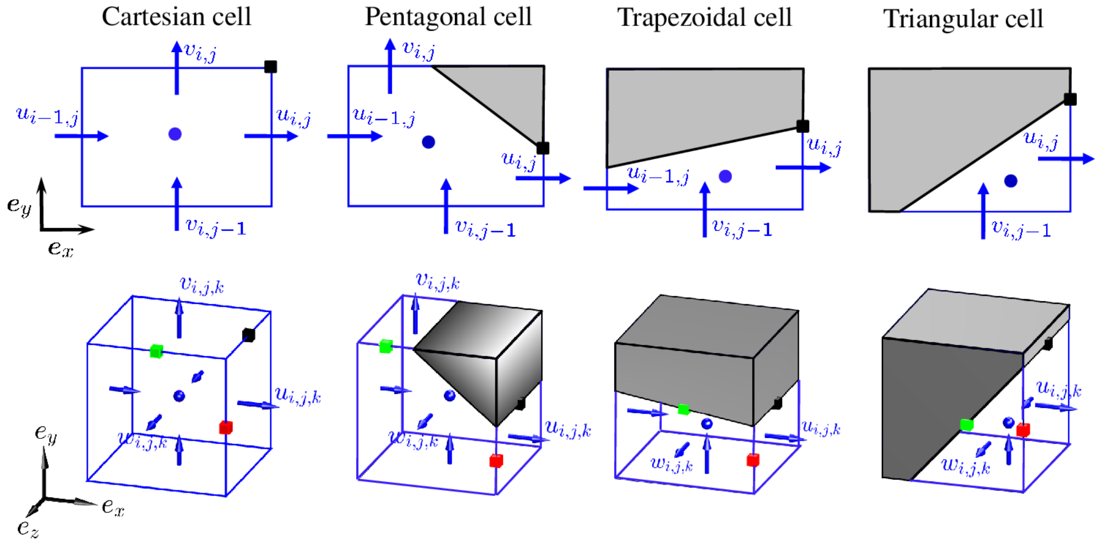

Figure 1: Cartesian cell and 3 basic types of cut-cells present in the LS-STAG mesh in 2D (top, from left to right) and 3D extruded (bottom) geometries. Velocity unknowns are located at face centroids. Pressure, diagonal components of $\nabla \boldsymbol{v}$ and temperature are located at cell centroids (●). Off-diagonal components of $\nabla \boldsymbol{v}$ are located at cell vertices in 2D, and edge centroids in 3D : $\frac{\partial u}{\partial y}$, $\frac{\partial v}{\partial x}$ (■), $\frac{\partial w}{\partial x}$, $\frac{\partial u}{\partial z}$ (🟥) and $\frac{\partial w}{\partial y}$, $\frac{\partial v}{\partial z}$ (🟩).

However, the discretization of cell-face dissipative fluxes involving shear stresses such as $\partial w/\partial x$ and $\partial w/\partial y$, which were absent from the 2D case, needed further attention. In effect, due to the non-orthogonality of the cut-cells, the use of 2-point formulas for computing these fluxes proves to be inaccurate. A way to improve the accuracy of the dissipative fluxes is to compute the whole velocity gradient at the cell faces, thus decomposing the flux as an orthogonal contribution (using a standard 2-point formula) and a non-orthogonal correction. This gradient reconstruction technique at cell faces is commonly denominated “secondary gradients” [5, 6, 7, 8, 9] in the CFD community and “diamond cell method” [10, 11, 12] in the applied mathematics community, since the discrete gradient can be derived by applying the Gauss formula in the diamond-shaped control-volume formed by neighboring cell centers and vertices. For this reason, this technique will subsequently be denominated “diamond cell method”, or simply DCM. In the DCM, the non-orthogonal component can be computed by a deferred correction algorithm using discrete gradients calculated at cell centers [7, 8, 9], or can be expressed from data at cell vertices [6, 10, 11, 12]. The data at cell vertices are then interpolated from the velocity unknowns and the boundary conditions, most notably by using least-squares reconstruction [6, 10]. The gradient reconstruction with vertex data is the one that is pursued in this paper, since it allows an implicit solution of linear systems with black-box iterative solvers [13]. Note also that in a Cartesian cut-cell mesh, non-orthogonal correction is only needed in the vicinity of the immersed boundary.

This non-orthogonality issue is also found for computing heat transfer problems with the 2D version of the LS-STAG method, where the temperature unknowns are positioned at cell centers. Hence, for the ease of discussion the diamond cell method will first be described and analyzed for the diffusive fluxes of the heat equation, using various interpolation schemes for vertex reconstruction. In addition to the interpolation schemes usually employed by unstructured methods (inverse distance weighting, least-squares, see the discussion in [12]), we will propose a barycenteric interpolation based on triangulation that produces excellent results on our cut-cell meshes. Simple and effective strategies for implementing Dirichlet and Neumann boundary conditions will also be presented. The accuracy of the DCM and the various variants of vertex reconstruction will be investigated on a series of benchmark problems (Taylor-Couette exact solution, natural convection from a cylinder in an enclosure [14]) by inspecting the formal order of accuracy and the heat flux distribution at the immersed boundary. In particular, the impact of the cut-cell quality (non-orthogonality, smoothness ratio) on the accuracy of the heat flow distribution will be thoroughly investigated. One of the main findings is that the largest errors in the diffusive flux computations are located in cut-cells of very small size (the so-called "small cut-cell problem" that hampers the stability and accuracy of cut-cell methods, see [15] for a review). The use of DCM with second-order vertex reconstruction greatly limits the magnitude of these local errors, and does not reduce the stability of the numerical solution, since DCM and 2-point results have been obtained with the same time-integration scheme and time-step values. In this respect, DCM gradient discretization can be considered as a viable alternative to *ad-hoc* procedures such as cell-merging [16] or cell-linking [17].

Finally, the diamond cell technique is employed for enhancing the accuracy of the dissipative terms for isothermal flows in 3D extruded geometries, most notably the shear stresses $\partial w/\partial x$ and $\partial w/\partial y$ which were discretized with 2-point formulas in [4] in spite of the non-orthogonality of the cut-cells. Furthermore, DCM formulas at the boundary enables us to revisit the discretization of the hydrodynamic forces at the immersed boundary, in order to provide a more systematic and accurate formulation than the one proposed in our previous works [1, 4]. This formulation is valid for the 2D and 3D-extruded version of the LS-STAG method. The method will be applied to the unsteady flow past circular cylinder [18] already computed in [4] with the 2-point LS-STAG method and an unstructured commercial solver (ANSYS FLUENT©), and the benefits of the DCM discretization in terms of accuracy and computational cost will be investigated.

2. Discretization of diffusion with the diamond cell technique

The various discretization of diffusion that we consider are best detailed for the 2D temperature equation :

$$\frac{\partial T}{\partial t} + \nabla \cdot (\boldsymbol{v} T) - \kappa \boldsymbol{\nabla}^2 T = 0, \tag{1}$$

where T is the temperature, $\boldsymbol{v} = (u, v)$ is the convection velocity and κ is the thermal diffusivity. In the cut-cell $\Omega_{i,j}$ depicted in red in Fig. 2, whose volume is denoted $V_{i,j}$, the temperature $T_{i,j}$ is located at the centroid $\boldsymbol{x}^{\mathrm{G}}_{i,j} = (x^G_{i,j}, y^G_{i,j})$, and the faces of the cut-cell are decomposed as $\Gamma_{i,j} = \Gamma^{\mathrm{w}}_{i,j} \cup \Gamma^{\mathrm{e}}_{i,j} \cup \Gamma^{\mathrm{s}}_{i,j} \cup \Gamma^{\mathrm{n}}_{i,j} \cup \Gamma^{\mathrm{ib}}_{i,j}$, where $\Gamma^{\mathrm{ib}}_{i,j}$ denotes the solid part of the cut-cell, while the fluid faces are denoted with the standard compass notations. The diffusive flux at the east face $F^{\mathrm{d}}_{\mathrm{e}} = \kappa \int_{\Gamma^{\mathrm{e}}_{i,j}} \nabla T \cdot \boldsymbol{n} \, \mathrm{d}S$ is discretized with midpoint quadrature, yielding :

$$F^{\mathrm{d}}_{\mathrm{e}} \cong \kappa \, \theta^u_{i,j} \Delta y_j \left. \frac{\partial T}{\partial x} \right|_{\mathrm{e}}, \tag{2}$$

where $\theta^u_{i,j} \in [0,1]$ is the fluid fraction ratio of the face, such as $\theta^u_{i,j}\Delta y_j = \|\boldsymbol{x}_{\mathrm{N}} - \boldsymbol{x}_{\mathrm{S}}\|$ in Fig. 2, and $\partial T/\partial x|_{\mathrm{e}} = \nabla T|_{\mathrm{e}} \cdot \boldsymbol{n}_{\mathrm{e}}$ is the face-normal temperature gradient.

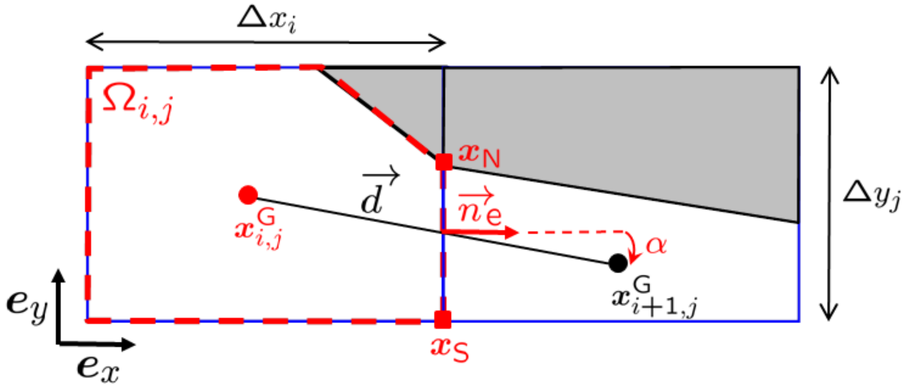

Figure 2: Relevant notations for the discretization of the diffusion flux at the east face of cut-cell $\Omega_{i,j}$.

Several alternate formulas can be considered for computing $\partial T/\partial x|_{\mathrm{e}}$. The simplest is the following two-point approximation :

$$\left. \frac{\partial T}{\partial x} \right|_{\mathrm{e}} \cong \frac{T_{i+1,j} - T_{i,j}}{\|\boldsymbol{d}\|}, \tag{3}$$

where $\boldsymbol{d} = \boldsymbol{x}^{\mathrm{G}}_{i+1,j} - \boldsymbol{x}^{\mathrm{G}}_{i,j}$ is the vector joining the cell centroids. In the case where $\boldsymbol{d}$ and the face normal vector $\boldsymbol{n}_{\mathrm{e}} = \boldsymbol{e}_x$ are colinear, formulae (2)-(3) recover the usual second-order Cartesian discretization. However, for the cut-cells depicted in Fig. 2, the colinearity is lost and formula (3) loses its accuracy as the non-colinearity increases, *i.e.* as the angle $\alpha = \mathrm{mes}(\boldsymbol{n}_{\mathrm{e}}, \boldsymbol{d})$ becomes large. This issue, which is analogous to the non-orthogonality of curvilinear and unstructured grids [9], has first been observed in [19] for the LS-STAG method in 3D extruded geometries. In this reference, they proposed to compute off-diagonal components of the velocity gradients of the Navier-Stokes equations with a formula consistent with the pressure gradient discretization. For the case of the temperature equation, this *ad hoc* formula amounts to :

$$\left. \frac{\partial T}{\partial x} \right|_{\mathrm{e}} \cong \frac{T_{i+1,j} - T_{i,j}}{(\frac{1}{2}V_{i+1,j} + \frac{1}{2}V_{i,j})/\theta^u_{i,j}\Delta y_j}. \tag{4}$$

However, the Navier-Stokes computations in [4] reported that the use of the above two-point formulas diminishes the accuracy of the LS-STAG method, and that only a superlinear order of convergence was obtained. The reason is that, when $\alpha \neq 0$, the face-normal gradient $\partial T/\partial x|_{\mathrm{e}}$ cannot be written as a function of $T_{i,j}$ and $T_{i+1,j}$ only, it also has a component tangential to the face. It is thus necessary to compute the whole temperature gradient, which reads in the local basis of the east face :

$$\nabla T|_{\mathrm{e}} = \left. \frac{\partial T}{\partial x} \right|_{\mathrm{e}} \boldsymbol{e}_x + \left. \frac{\partial T}{\partial y} \right|_{\mathrm{e}} \boldsymbol{e}_y, \tag{5}$$

since $\boldsymbol{n}_{\mathrm{e}} = \boldsymbol{e}_x$ in Fig. 2.

An elegant way to proceed is to approximate the gradient in the $\boldsymbol{d}$ direction as :

$$\nabla T|_{\mathrm{e}} \cdot \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|} = \cos\alpha \left.\frac{\partial T}{\partial x}\right|_{\mathrm{e}} + \sin\alpha \left.\frac{\partial T}{\partial y}\right|_{\mathrm{e}} \cong \frac{T_{i+1,j} - T_{i,j}}{\|\boldsymbol{d}\|}, \tag{6}$$

while the gradient tangential to the face is computed as :

$$\left.\frac{\partial T}{\partial y}\right|_{\mathrm{e}} \cong \frac{T_{\mathrm{N}} - T_{\mathrm{S}}}{\theta^u_{i,j}\Delta y_j}, \tag{7}$$

where T_{N} and T_{S} are located at the face endpoints $\boldsymbol{x}_{\mathrm{N}}$ and $\boldsymbol{x}_{\mathrm{S}}$. Then, substitution of (7) in (6) gives the following approximation of the face-normal gradient :

$$\left.\frac{\partial T}{\partial x}\right|_{\mathrm{e}} \cong \frac{T_{i+1,j} - T_{i,j}}{\|\boldsymbol{d}\|\cos\alpha} - \tan\alpha \frac{T_{\mathrm{N}} - T_{\mathrm{S}}}{\theta^u_{i,j}\Delta y_j}, \tag{8a}$$

with

$$\|\boldsymbol{d}\|\cos\alpha = x^{\mathrm{G}}_{i+1,j} - x^{\mathrm{G}}_{i,j}, \quad \tan\alpha = \frac{y^{\mathrm{G}}_{i+1,j} - y^{\mathrm{G}}_{i,j}}{x^{\mathrm{G}}_{i+1,j} - x^{\mathrm{G}}_{i,j}}. \tag{8b}$$

The first quotient in the RHS of (8a) is the orthogonal contribution of the gradient discretization, while the second quotient is the non-orthogonal contribution that vanishes when $\boldsymbol{d}$ and $\boldsymbol{n}_{\mathrm{e}}$ are colinear ($\alpha = 0$), yielding a two-point approximation identical to (3) and (4). Compared to the two-point approximations, this formula introduces supplementary unknowns at face vertices $\boldsymbol{x}_{\mathrm{N}}$ and $\boldsymbol{x}_{\mathrm{S}}$, that need to be reconstructed for the solution at the centroids, as discussed in the next section.

The gradient formula (8) can also be derived by applying the Green-Gauss theorem in the diamond-shaped control volume formed by joining the cell centers $\boldsymbol{x}^{\mathrm{G}}_{i,j}$, $\boldsymbol{x}^{\mathrm{G}}_{i+1,j}$ and the face endpoints $\boldsymbol{x}_{\mathrm{N}}$, $\boldsymbol{x}_{\mathrm{S}}$; the resulting formula is known as the *diamond cell scheme*. The diamond cell method (hereafter labeled DCM) has been introduced in [20, 5, 6] for viscous compressible computations on unstructured meshes. Mathematical analysis has been performed in [10, 11] for the convection-diffusion equation. Furthermore, formula (8) is equivalent to the over-relaxed approach for diffusion discretization presented in [9].

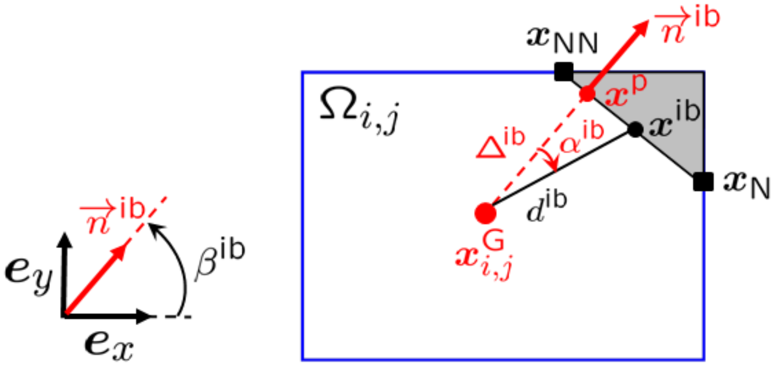

Figure 3: Relevant notations for the discretization of the diffusion flux at the solid face of cut-cell $\Omega_{i,j}$. For sake of clarity, some of the subscripts i, j have been omitted in the figure.

A similar procedure is employed to approximate diffusion at the immersed boundary. The diffusion flux at the immersed boundary is $F^{\mathrm{d}}_{\mathrm{ib}} \equiv \kappa \int_{\Gamma^{\mathrm{ib}}_{i,j}} \nabla T \cdot \boldsymbol{n}\, \mathrm{d}S \cong \kappa \, \|\boldsymbol{\Delta S}^{\mathrm{ib}}_{i,j}\| \; \partial T/\partial n|_{\mathrm{ib}}$, where $\boldsymbol{\Delta S}^{\mathrm{ib}}_{i,j}$ is the face area vector of the solid face whose coordinates are given by :

$$[\Delta S_x]^{\mathrm{ib}}_{i,j} = (\theta^u_{i-1,j} - \theta^u_{i,j})\Delta y_j, \qquad [\Delta S_y]^{\mathrm{ib}}_{i,j} = (\theta^v_{i,j-1} - \theta^v_{i,j})\Delta x_i, \tag{9}$$

for all types of cut-cells. Application of the DCM technique (6)-(8) in the direction $d^{\mathrm{ib}}_{i,j}$ joining centroid $x^{\mathrm{G}}_{i,j}$ to the center $\boldsymbol{x}^{\mathrm{ib}}_{i,j}$ of the solid face $\Gamma^{\mathrm{ib}}_{i,j}$ (see Fig. 3) yields the following gradient approximation in the local basis $(\boldsymbol{n}^{\mathrm{ib}}_{i,j},\, \boldsymbol{t}^{\mathrm{ib}}_{i,j})$ of the face :

$$\left.\frac{\partial T}{\partial n}\right|_{\mathrm{ib}} \cong \frac{T^{\mathrm{ib}}_{i,j} - T_{i,j}}{\Delta^{\mathrm{ib}}_{i,j}} - \tan\alpha^{\mathrm{ib}}_{i,j} \frac{T_{\mathrm{NN}} - T_{\mathrm{N}}}{\|\boldsymbol{x}_{\mathrm{NN}} - \boldsymbol{x}_{\mathrm{N}}\|}, \tag{10a}$$

$$\left.\frac{\partial T}{\partial t}\right|_{\mathrm{ib}} \cong \frac{T_{\mathrm{NN}} - T_{\mathrm{N}}}{\|\boldsymbol{x}_{\mathrm{NN}} - \boldsymbol{x}_{\mathrm{N}}\|}, \tag{10b}$$

where $T^{\mathrm{ib}}_{i,j}$ is the temperature at $\boldsymbol{x}^{\mathrm{ib}}_{i,j}$ and $\Delta^{\mathrm{ib}}_{i,j} = d^{\mathrm{ib}}_{i,j} \cos\alpha^{\mathrm{ib}}_{i,j}$ is the perpendicular distance from centroid $\boldsymbol{x}^{\mathrm{G}}_{i,j}$ to segment $\Gamma^{\mathrm{ib}}_{i,j}$. For some applications (see Sect 4.3), the components of the gradient at the solid boundary are needed in the Cartesian basis $(\boldsymbol{e}_x,\, \boldsymbol{e}_y)$ of the computational domain. By denoting $\beta^{\mathrm{ib}}_{i,j} = \mathrm{mes}(\boldsymbol{e}_x,\, \boldsymbol{n}^{\mathrm{ib}}_{i,j})$ the inclination of the solid face, a simple change of basis yields :

$$\left.\frac{\partial T}{\partial x}\right|_{\mathrm{ib}} \cong \cos\beta^{\mathrm{ib}}_{i,j} \frac{T^{\mathrm{ib}}_{i,j} - T_{i,j}}{\Delta^{\mathrm{ib}}_{i,j}} - \left(\sin\beta^{\mathrm{ib}}_{i,j} + \cos\beta^{\mathrm{ib}}_{i,j} \tan\alpha^{\mathrm{ib}}_{i,j}\right) \frac{T_{\mathrm{NN}} - T_{\mathrm{N}}}{\|\boldsymbol{x}_{\mathrm{NN}} - \boldsymbol{x}_{\mathrm{N}}\|}, \tag{11a}$$

$$\left.\frac{\partial T}{\partial y}\right|_{\mathrm{ib}} \cong \sin\beta^{\mathrm{ib}}_{i,j} \frac{T^{\mathrm{ib}}_{i,j} - T_{i,j}}{\Delta^{\mathrm{ib}}_{i,j}} + \left(\cos\beta^{\mathrm{ib}}_{i,j} - \sin\beta^{\mathrm{ib}}_{i,j} \tan\alpha^{\mathrm{ib}}_{i,j}\right) \frac{T_{\mathrm{NN}} - T_{\mathrm{N}}}{\|\boldsymbol{x}_{\mathrm{NN}} - \boldsymbol{x}_{\mathrm{N}}\|}, \tag{11b}$$

where the boundary inclination is efficiently calculated as $\cos\beta^{\mathrm{ib}}_{i,j} = [\Delta S_x]^{\mathrm{ib}}_{i,j}/\|\boldsymbol{\Delta S}^{\mathrm{ib}}_{i,j}\|$ and $\sin\beta^{\mathrm{ib}}_{i,j} = [\Delta S_y]^{\mathrm{ib}}_{i,j}/\|\boldsymbol{\Delta S}^{\mathrm{ib}}_{i,j}\|$.

3. Practical implementation of the diamond cell scheme

In the DCM formulae (8a) and (10), solution unknowns at the vertices of the solid boundary ($\boldsymbol{x}_{\mathrm{N}}$ and $\boldsymbol{x}_{\mathrm{NN}}$) and in the fluid domain ($\boldsymbol{x}_{\mathrm{S}}$) need to be interpolated from the boundary conditions and from neighboring cell-centered values in the fluid domain. Boundary data are discretized at the endpoints of the solid segments representing the immersed boundary, e.g. $\boldsymbol{x}_{\mathrm{N}}$ and $\boldsymbol{x}_{\mathrm{NN}}$ in Fig. 3.

3.1. Implementation of the gradient formulas

In the DCM formulas (8), all coefficients are efficiently calculated with explicit formulas. In formulas (10) and (11), the computation of $\Delta^{\mathrm{ib}}_{i,j}$ and $\tan\alpha^{\mathrm{ib}}_{i,j}$ is performed after having calculated $\boldsymbol{x}^{\mathrm{P}}_{i,j}$, the orthogonal projection of $\boldsymbol{x}^{\mathrm{G}}_{i,j}$ on the solid face $\Gamma^{\mathrm{ib}}_{i,j}$ (see Fig. 2). Then we get $\Delta^{\mathrm{ib}}_{i,j} = \|\boldsymbol{x}^{\mathrm{P}}_{i,j} - \boldsymbol{x}^{\mathrm{G}}_{i,j}\|$ and :

$$\tan\alpha^{\mathrm{ib}}_{i,j} = \frac{x^{\mathrm{ib}}_{i,j} - x^{\mathrm{P}}_{i,j}}{\Delta^{\mathrm{ib}}_{i,j}}, \tag{12}$$

where $x^{\mathrm{ib}}_{i,j}$ and $x^{\mathrm{P}}_{i,j}$ denote the abscissae of $\boldsymbol{x}^{\mathrm{ib}}_{i,j}$ and $\boldsymbol{x}^{\mathrm{P}}_{i,j}$ in the interval $(x_{\mathrm{N}}, x_{\mathrm{NN}})$.

As a matter of fact, the DCM formula (10) (or (11)) at the boundary is independent of the point where it is calculated when the data at the immersed boundary is assumed to vary linearly. This can be shown by writing the abscissa of $\boldsymbol{x}_{i,j}^{\mathrm{ib}}$ as :

$$x_{i,j}^{\mathrm{ib}} = \omega_{\mathrm{ib}} x_{\mathrm{NN}} + (1-\omega_{\mathrm{ib}}) x_{\mathrm{N}}, \quad \omega_{\mathrm{ib}} = \frac{x_{i,j}^{\mathrm{ib}} - x_{\mathrm{N}}}{x_{\mathrm{NN}} - x_{\mathrm{N}}}, \tag{13a}$$

and assuming linear variation of the boundary values :

$$T_{i,j}^{\mathrm{ib}} \cong \omega_{\mathrm{ib}} T_{\mathrm{NN}} + (1-\omega_{\mathrm{ib}}) T_{\mathrm{N}}. \tag{13b}$$

After inserting (12) and (13b) in (10a), one gets :

$$\left.\frac{\partial T}{\partial n}\right|_{\mathrm{ib}} \cong \frac{\omega_{\mathrm{P}} T_{\mathrm{NN}} + (1-\omega_{\mathrm{P}}) T_{\mathrm{N}} - T_{i,j}}{\Delta_{i,j}^{\mathrm{ib}}}, \quad \omega_{\mathrm{P}} = \frac{x_{i,j}^{\mathrm{P}} - x_{\mathrm{N}}}{x_{\mathrm{NN}} - x_{\mathrm{N}}}, \tag{14}$$

which corresponds to the discretization of the temperature gradient in the direction normal to the boundary, since $\omega_{\mathrm{P}} T_{\mathrm{NN}} + (1-\omega_{\mathrm{P}}) T_{\mathrm{N}}$ is a linear approximation of the temperature at $\boldsymbol{x}_{i,j}^{\mathrm{P}}$, denoted $T_{i,j}^{\mathrm{P}}$. This property shows that, for non-homogeneous boundary data, the discrete gradient can be calculated at any location on the solid boundary. This property will prove useful for the practical implementation of the DCM method, most notably for the Neumann boundary conditions discussed next, and the calculation of the hydrodynamic forces discussed in Sect. 4.3.

3.2. Implementation of boundary conditions

In the case of Dirichlet boundary conditions, the temperature is prescribed at the endpoints of the solid faces, providing thus the values of T_{N}, T_{NN} and $T_{i,j}^{\mathrm{ib}} = (T_{\mathrm{N}} + T_{\mathrm{NN}})/2$ in (8a) and (10a). Note that in the case of homogeneous boundary conditions, the non-orthogonal component in (10a) vanishes.

For the Neumann boundary condition $\partial T/\partial n|_{\mathrm{ib}} = \phi$, the flux at the solid boundary is explicitly given by $F_{\mathrm{ib}}^{\mathrm{d}} \cong \kappa \|\boldsymbol{\Delta S}_{i,j}^{\mathrm{ib}}\| \phi$ and the value of T_{N} in (8a) is reconstructed from the following procedure. First, linear variation of the temperature at the boundary yields :

$$T_{\mathrm{N}} \cong \omega T_{i,j}^{\mathrm{p}} + (1-\omega) T_{i+1,j}^{\mathrm{p}}, \quad \omega = \frac{\|\boldsymbol{x}_{i+1,j}^{\mathrm{p}} - \boldsymbol{x}_{\mathrm{N}}\|}{\|\boldsymbol{x}_{i+1,j}^{\mathrm{p}} - \boldsymbol{x}_{\mathrm{N}}\| + \|\boldsymbol{x}_{i,j}^{\mathrm{p}} - \boldsymbol{x}_{\mathrm{N}}\|}. \tag{15}$$

Next, $T_{i,j}^{\mathrm{p}}$ is reconstructed from cell-centered values by using the boundary condition, yielding $T_{i,j}^{\mathrm{p}} \cong T_{i,j} + \phi \Delta_{i,j}^{\mathrm{ib}}$ from (14). After using a similar approximation for $T_{i+1,j}^{\mathrm{p}}$, the value of T_{N} is finally given by :

$$T_{\mathrm{N}} \cong \omega\big(\phi \Delta_{i,j}^{\mathrm{ib}} + T_{i,j}\big) + (1-\omega)\big(\phi \Delta_{i+1,j}^{\mathrm{ib}} + T_{i+1,j}\big). \tag{16}$$

3.3. Solution reconstruction at fluid vertices

In the DCM formula (8a), the temperature at $\boldsymbol{x}_{\mathrm{S}}$ needs to be reconstructed from neighboring cell-centered values :

$$T_{\mathrm{S}} \cong \sum_{(i,j) \in N_{\mathrm{b}}(\mathrm{S})} \omega_{i,j} T_{i,j}, \tag{17}$$

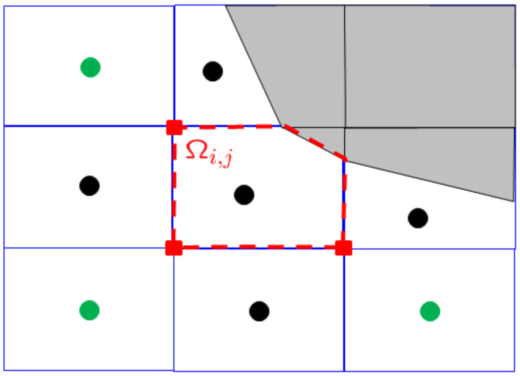

Figure 4: Stencil used for the discretization of diffusion in cut-cell $\Omega_{i,j}$. In black: 5-point stencil for 2-point discretization. For the DCM discretization with reconstruction at vertices (🟥), unknowns at the green centroids are added to the stencil.

where $N_{\mathrm{b}}(\mathrm{S})$ refers to the indices of the 4 cells neighboring vertex $\boldsymbol{x}_{\mathrm{S}}$. Bertolazzi & Manzini [12] reviewed various possibilities for defining the reconstruction weights $\omega_{i,j}$. Basic choices include the inverse-volume interpolation, with weights based on the cell volumes :

$$\omega_{i,j} = \frac{V_{i,j}}{\displaystyle\sum_{(i,j)\in N_{\mathrm{b}}(\mathrm{S})} V_{i,j}}, \tag{18}$$

and the inverse-distance interpolation, with weights :

$$\omega_{i,j} = \frac{\|\boldsymbol{x}^{\mathrm{G}}_{i,j} - \boldsymbol{x}_{\mathrm{S}}\|^{-p}}{\displaystyle\sum_{(i,j)\in N_{\mathrm{b}}(\mathrm{S})} \|\boldsymbol{x}^{\mathrm{G}}_{i,j} - \boldsymbol{x}_{\mathrm{S}}\|^{-p}}, \tag{19}$$

where $p \geq 1$ is a given exponent ($p = 1$ corresponds to the Euclidian distance).

These two choices of weights give exact reconstruction for constant functions only, reducing thus the formal accuracy of the DCM formula to first-order only. In [12], an algorithm is discussed to reach second-order accuracy, by altering the values of the weights in order to fulfill the condition for exact reconstruction of linear functions. Note however that this algorithm involves centroid selection, solution of linear systems, and the resulting set of weights is not unique.

Second-order accuracy can also be reached by using least-squares weights that minimize the reconstruction of linear functions. This least-squares reconstruction is described in [6] for the solution of the compressible Navier-Stokes equations on unstructured grids, and has been analyzed in [10, 12] for model diffusion problems. However, our numerical experiments (see next section) showed that negative weights may appear for some cut-cells, giving rise to oscillatory solutions. For this reason, we have implemented a reconstruction based on triangulation [21] that ensures non-negativity of the weights. Out of the 4 triangles formed by the neighboring centroids of a given vertex, we select the triangle of largest minimal angle that contains the vertex. The value at the vertex is then obtained by barycentric interpolation.

3.4. Other implementation issues

In the LS-STAG code, diffusion is treated with an implicit time-integration method, so the use of the DCM with vertex reconstruction has the effect to increase the stencil of the linear systems. As shown in Fig. 4, the

DCM discretization needs to be applied to the four fluid faces of cut-cell $\Omega_{i,j}$, yielding a 9-point stencil instead of the standard 5-point stencil for 2-point discretization. Note that DCM needs only to be applied to neighboring cells of the immersed boundary. In the vast majority of the computational domain, the mesh is orthogonal and the standard 5-point stencil is recovered. The linear systems are solved with the HYPRE library (V2.10) using the semi-structured-grid system interface [13].

4. Numerical results

The diamond cell method is implemented in the 2D and 3D-extruded LS-STAG code for the discretization of face-normal gradients in the temperature and Navier-Stokes equations. First, an accuracy study of the DCM is presented for the 2D temperature equation. The temperature equation is then coupled with the Navier-Stokes equations for computing 2D natural convection problems, with the aim to compute accurately local heat flux at immersed boundaries, which is a sensible quantity since it involves temperature gradients at the solid face of cut-cells. Once the DCM is assessed on 2D flows, the DCM is applied to the Navier-Stokes equations in 3D-extruded geometries to enhance the accuracy of isothermal incompressible flows computed with the LS-STAG method with 2-point discretization reported in [4].

For interpolation at fluid vertices, the following reconstruction schemes discussed in Sect. 3.3 are considered :

– LS-STAG/DCM-ID1: first-order inverse distance weights with $p = 2$,

– LS-STAG/DCM-ID2: second-order inverse distance weights with $p = 2$, using the algorithm discussed in [12],

– LS-STAG/DCM-LSQ: least-squares weights,

– LS-STAG/DCM-TRI: barycentric interpolation on triangulation.

For all variants of the DCM, Dirichlet and Neumann boundary conditions are imposed following the procedures discussed in Sect. 3.2. The performance of DCM is also compared with the 2-point gradient (4), which is labeled LS-STAG/2PT.

All computations are performed with the parallel LS-STAG flow solver described in [4]. Unsteady results are obtained with second-order time accurate time stepping, while steady state results are obtained with first-order time stepping, and is considered reached when the normalized time derivatives of the velocity and temperature is below the threshold $\epsilon = 10^{-7}$ (for the 2D flows of Sections 4.1 and 4.2), or $\epsilon = 10^{-6}$ (for the 3D extruded flow of Section 4.3). For the natural convection benchmark of Sect. 4.2, an additional convergence condition based on the wall heat flux balance is considered.

4.1. 2D Thermal conduction in Taylor-Couette cell

First the spatial accuracy of the various variants of DCM is assessed on the test case of pure thermal conduction between concentric cylinders of diameters $r = R_1$ and $r = R_2$, for which the steady radial temperature distribution :

$$T_{\text{ex}}(r) = \phi R_1 \ln(r/R_2) + T_2, \quad \phi = \frac{T_1 - T_2}{R_1 \ln(R_1/R_2)}, \tag{20}$$

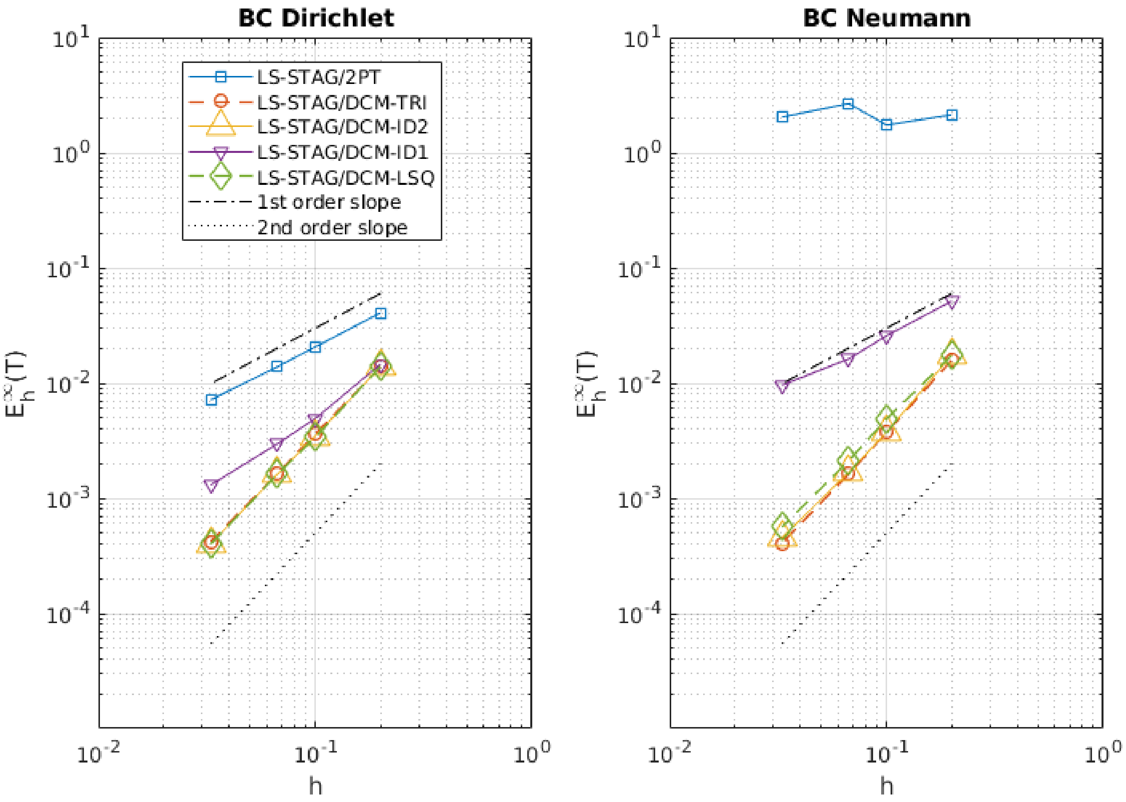

Figure 5: Maximal error on the temperature versus grid size h for the pure conduction test case for both types of boundary conditions.

is solution to the heat equation (Eq (1) with $\boldsymbol{v} = 0$) with both Dirichlet ($T_{\text{ex}}(R_1) = T_1$, $T_{\text{ex}}(R_2) = T_2$ for $r = R_2$) or Neumann ($\partial T_{\text{ex}}(R_1)/\partial r = \phi$, $\partial T_{\text{ex}}(R_2)/\partial r = \phi R_1/R_2$) boundary conditions. For the numerical experiments, we set the temperature difference as $T_1 - T_2 = 1\text{K}$, and employ the same geometry ($R_1 = 1\,\text{m}$ and $R_2 = 4\,\text{m}$) and uniform meshes (of size $\Delta x = \Delta y = h$) considered in [2] for non-Newtonian computations with the LS-STAG method.

For both Dirichlet and Neumann boundary conditions, Fig. 5 compares the maximal error :

$$E_h^\infty(T) = \frac{\max\limits_{\text{CVs } \Omega_{i,j}} \left| T_{i,j} - T_{\text{ex}}(\boldsymbol{x}_{i,j}^{\text{G}}) \right|}{\max\limits_{\text{CVs } \Omega_{i,j}} \left| T_{\text{ex}}(\boldsymbol{x}_{i,j}^{\text{G}}) \right|} \tag{21}$$

obtained on the whole computational domain (including the cut-cells) for the various DCM variants. The 2-point method gives clearly the worst results, with a first-order rate of convergence for the Dirichlet case while no convergence is observed for the Neumann case. The errors given by all DCM variants are several decades lower, highlighting the benefits of this gradient computation. Note however that the errors of DCM-ID1 quickly saturates to a first-order rate when the grid size is reduced, due to the low accuracy of the vertex reconstruction scheme. All other reconstruction schemes (DCM-ID2, DCM-LSQ, DCM-TRI) give a similar second-order accuracy, with almost indistinguishable results on this test case. The performance of these three variants will be more thoroughly investigated in the next section.

4.2. 2D Natural convection from a cylinder in an enclosure

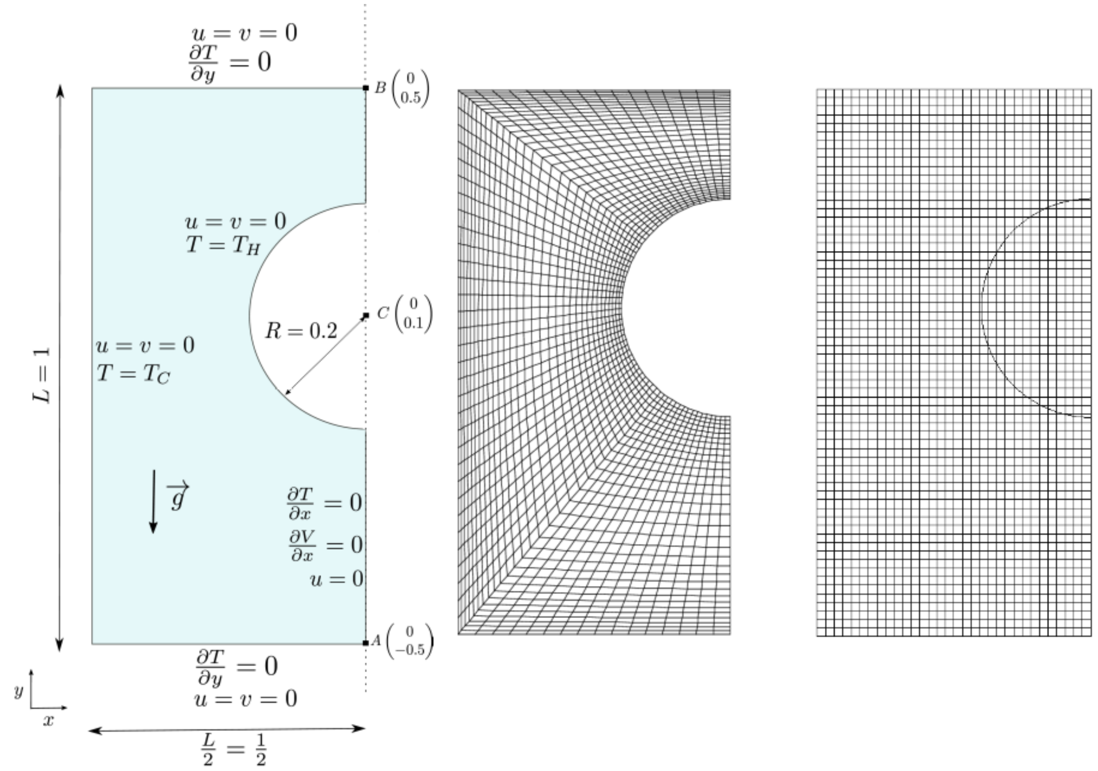

Figure 6: Sketch of the computational domain for the natural convection test-case (with $L = 1\,\mathrm{m}$, $T_{\mathrm{H}} = 1\,\mathrm{K}, T_{\mathrm{C}} = 0\,\mathrm{K}$), curvilinear mesh L1 and Cartesian mesh M1 of size 32×64 used respectively for the FLUENT© and LS-STAG computations.

The second test-case concerns a fluid flow and heat transfer problem in complex geometries, for which Demirdžić *et al* [14] published accurate benchmark solutions. This test case concerns natural convection of a heated cylinder placed eccentrically in an inclosure (test C4 in [14]), where the heat equation (1) is coupled with the Navier-Stokes equations in the Boussinesq approximation. The computational domain and boundary conditions are shown in Fig. 6 (left), with gravity acting in the downwards direction; since the solution possesses a vertical symmetry at $x = 0$, only half of the geometry is considered. As in [14], the dimensions and fluid properties are: enclosure length $L = 1\,\mathrm{m}$, density $\rho = 1\,\mathrm{kg{\cdot}m^{-3}}$, gravity constant $g = 1\,\mathrm{m{\cdot}s^{-2}}$, thermal expansion coefficient $\beta = 0.1\mathrm{K}^{-1}$, thermal diffusivity $\kappa = 10^{-4}\,\mathrm{m}^2 \cdot \mathrm{s}^{-1}$, dynamic viscosity $\mu = 10^{-3}\,\mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-1}$ and temperature difference $T_{\mathrm{H}} - T_{\mathrm{C}} = 1\,\mathrm{K}$, that set the Rayleigh and Prandtl number respectively as:

$$\mathrm{Ra} \equiv \frac{\beta \rho g L^3}{\mu \kappa}(T_{\mathrm{H}} - T_{\mathrm{C}}) = 10^6, \tag{22}$$

$$\mathrm{Pr} \equiv \frac{\mu}{\rho \kappa} = 10. \tag{23}$$

The purpose of these computations is to assess the ability of the LS-STAG/DCM method to compute accurately gradients at the immersed boundary, in the present case the average Nusselt number on the half-cylinder of length $|\Gamma^{\mathrm{ib}}|$:

$$\overline{\mathrm{Nu}} = \frac{1}{|\Gamma^{\mathrm{ib}}|\,(T_{\mathrm{H}} - T_{\mathrm{C}})/L} \int_{\Gamma^{\mathrm{ib}}} \nabla T \cdot \boldsymbol{n}\,\mathrm{d}S, \tag{24}$$

and the distribution of the local Nusselt number along the half cylinder, which is calculated at the center of the solid boundary of the cut-cells using formula (10a).

The LS-STAG computations are performed on a series of refined meshes M1-M5 of square uniform Cartesian cells, where the coarsest mesh M1 represented in Fig. 6 (right) has 32×64 cells and the finer meshes are obtained by doubling the number of cells in each direction, such as the finest mesh M5 has 512×1024 cells. For comparison purpose, we have also computed solutions with the commercial CFD package FLUENT© (V15.07), on a series of high-quality curvilinear meshes L1-L5 (see Fig. 6 (middle)) of same number of cells, with cell clustering near the heated cylinder. The FLUENT© computations have been performed with the pressure-based solver and the steady SIMPLE algorithm, using second-order spatial discretizations: QUICK scheme for convection discretization, Green-Gauss node-based interpolation for gradient calculation, and body-force weighted pressure interpolation. As it is done for the benchmark computations of [14], both LS-STAG and FLUENT© results are reported after the steady state is reached and the heat flux difference between the hot and cold walls is sufficiently low to give a four-digit accuracy on the average Nusselt number.

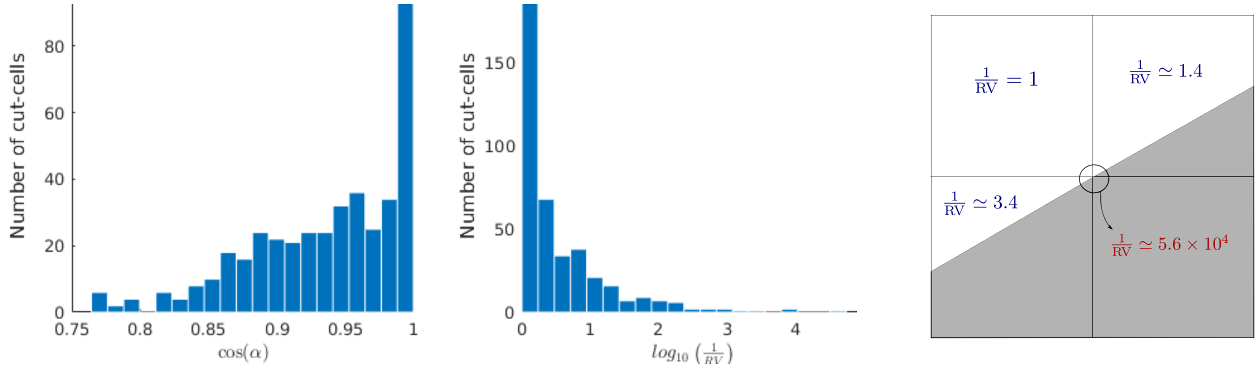

Figure 7: Histogram of the orthogonal quality (left) and volume-change ratio (middle) of the cut-cell M4 mesh; on this mesh the total number of cut-cells is 410. At right: mesh close-up in the vicinity of the cut-cell with the highest value of $1/\mathrm{RV}_{i,j}$.

It is worthwhile to inspect the quality of the meshes in the area of interest, *i.e.* the vicinity of the half cylinder. In effect, due to the Cartesian nature of the cut-cell mesh, it is very cumbersome to control the shape of the cut-cells, which may impact the accuracy of the discretization. Mesh quality is usually measured by two metrics : the orthogonal quality and the smoothness. The orthogonal quality $\mathrm{OQ}_{i,j}$ of cut-cell $\Omega_{i,j}$ is defined as the minimal value of $\cos\alpha$ in (8) for all its faces, where a value close to 0 corresponds to low quality cells. The mesh smoothness is measured by the volume-change ratio of the cut-cells :

$$\mathrm{RV}_{i,j}=\min_{(k,l)\in N_\mathrm{b}}\frac{V_{i,j}}{V_{k,l}}, \tag{25}$$

where N_b refers to the neighboring cells that share a face with cut-cell $\Omega_{i,j}$ (solid cells are excluded). For each cut-cell $\mathrm{RV}_{i,j}\in\,]0,1]$ since at least one neighboring cell is Cartesian, and a value close to 0 indicates a cut-cell of small volume compared to Cartesian cells. Fig. 7 reports both metrics for mesh M4 (similar trends are found on all cut-cell meshes). First, it can be observed that no cut-cells display low non-orthogonality values : it can be shown that for background Cartesian cells of size $\Delta x\times\Delta y$, the minimal value of $\mathrm{OQ}_{i,j}$ is $\min(\Delta x/\sqrt{\Delta x^2+\Delta y^2},\Delta y/\sqrt{\Delta x^2+\Delta y^2})$, which gives a minimal value of $\sqrt{2}/2$ for square cells consistent with the results of the figure. Nevertheless, the orthogonal quality of the curvilinear meshes is superior, with OQ $\simeq$ 0.95 near the half-cylinder. Concerning the smoothness metric, we observe that a small portion of the cut-cells display very low value of RV, indicating large changes of volume: 5% of the cut cells of meshes M1-M5 are such that $1/\mathrm{RV}_{i,j} > 100$. For comparison, the maximal value of 1/RV is 1.25 for the curvilinear meshes. As shown in Fig. 7 (right), which displays the M4 mesh in the vicinity of the most degenerate cut-cell, these cut-cells correspond to very small triangular cells surrounded by Cartesian and large pentagonal or trapezoidal cells. Hence, the mesh smoothness metric $\mathrm{RV}_{i,j}$ is an indicator of the occurence of the "small cut-cell problem" that hampers the stability and accuracy of most cut-cell methods (see [16, 17] and the review in [15]). In the following, we will investigate the local errors in heat flux distribution as a function of the cut-cell smoothness to evaluate the accuracy of the various gradient discretizations.

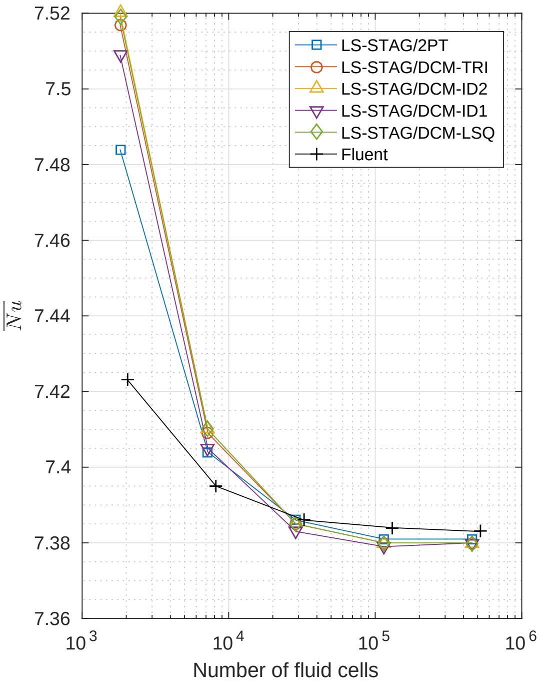

Figure 8: Convergence of the average Nusselt number with respect to the number of fluid cells for meshes M1/L1 to M5/L5.

Fig. 8 shows the grid convergence of the average Nusselt number, which is given in [14] with four significant digits as $\overline{\mathrm{Nu}} \simeq 7.384$. Irrespective of the gradient discretization scheme, all LS-STAG variants converge at a similar rate to the same grid-independent value $\overline{\mathrm{Nu}} \simeq 7.380$, while the FLUENT© computation yields $\overline{\mathrm{Nu}} \simeq$ 7.383 on the finest mesh. It is also worthwhile to observe that on the 2 coarsest meshes, the FLUENT© computations are more accurate than LS-STAG. The slight superiority of the FLUENT© results is certainly due to the higher quality of the curvilinear meshes near the complex boundary, as discussed above.

Fig. 9 (left) displays the distribution of the local Nusselt number along the immersed cylinder obtained for the LS-STAG and FLUENT© computations on the M4/L4 mesh, along with reference results [14]. In order to quantify the influence of the cut-cells shape on the accuracy of the results, the LS-STAG profiles are colorized with the volume-change ratio of the cut-cells (25). First it can be observed that the LS-STAG method with 2-point gradients gives a highly oscillatory profile, confirming the inaccuracy of this approximation of diffusion.

The use of the diamond scheme diminishes the level of oscillations for the first-order method DCM-ID1 and for DCM-LSQ. However, both methods still present oscillations in cut-cells with small volume-change ratio $\mathrm{RV}_{i,j}$, indicating inaccurate discretization in small cut-cells. In contrast, the profiles obtained with the other second-order interpolations (DCM-ID2 and DCM-TRI) are virtually free of any oscillations, and perfectly match the reference results. Fig. 9 (right) plots the local error on the Nusselt distribution with respect to the volume change of the cut-cells. For the 2PT, DCM-ID1 and DCM-LSQ discretization, the errors grow exponentially with 1/RV, showing clearly that small cut-cells are the source of the discretization errors. In contrast, it is striking to observe that for the most accurate discretizations (DCM-ID2 and DCM-TRI), the errors are virtually independent of the magnitude of RV. These results illustrate the robustness of these interpolations, which yields accurate discretization irrespective of the size and shape of the cut cells.

	M_1	M_2	M_3	M_4	M_5
LS-STAG/2PT	80.10	163.6	364.2	671.5	1345
LS-STAG/DCM-ID1	15.44	28.70	67.21	119.9	245.5
LS-STAG/DCM-ID2	9.430	8.440	8.837	8.179	8.291
LS-STAG/DCM-LSQ	14.35	13.12	15.33	13.89	20.60
LS-STAG/DCM-TRI	8.931	7.798	8.248	7.606	7.724

Table 1: Total variation (26) of the local Nusselt number for the various gradient discretizations on the M1-M5 grids.

Finally, in order to give a quantitative measure of the oscillations generated by the various gradient discretizations, Table 1 compares the total variation (TV) of the local Nusselt number at the cylinder surface :

$$\mathrm{TV} = \frac{1}{N_{\mathrm{cc}}} \sum_{i=1}^{N_{\mathrm{cc}}} \left| \frac{\mathrm{Nu}_{i+1} - \mathrm{Nu}_i}{\theta_{i+1} - \theta_i} \right|, \tag{26}$$

where N_{cc} is the number of cut-cells on the half cylinder. On a given grid distribution, the numerical solution with the lowest TV is the least oscillatory solution. This table confirms that on all grids DCM-ID2 and DCM-TRI are by far the most accurate methods, with a value of TV slightly inferior for the DCM-TRI interpolation.

As a conclusion, these computations show that accurate solutions can be achieved with the DCM discretization. Nonetheless, the accuracy of the DCM approximation is highly dependent on the interpolation scheme that is used, otherwise inaccurate results are encountered in the smaller cut-cells. These results show that the second-order methods DCM-ID2 and DCM-TRI stand as the best discretization methods, yielding almost indistinguishable results on the local Nusselt distribution. Owing to the simplicity of the triangular interpolation compared to the inverse distance interpolation of second order, the DCM-TRI variant is selected for implementation in the 3D-extruded code.

4.3. Flows in 3D extruded geometries

The DCM is now applied to the Navier-Stokes equations in 3D-extruded geometries to enhance the accuracy of the viscous fluxes computed with the LS-STAG method with 2-point discretization reported in [4]. In the

3 basic types of cut-cells shown in Fig. 1, the velocities unknowns are located at face centroids, the pressure and diagonal components of the rate-of-strain tensor $\dot{\boldsymbol{\gamma}} = \nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T$ are located at cell centroids, and the off-diagonal components of $\dot{\boldsymbol{\gamma}}$ are located at edge centroids. A complete description of the LS-STAG method in 3D-extruded geometries is given in [4].

Relevant notations for the DCM discretization are given in Fig. 10. Due to the extrusion in the z direction, the DCM discretization only needs to be applied to the off-diagonal components of ∇w (namely $\partial w/\partial x$ and $\partial w/\partial y$), while the off-diagonal components of ∇u and ∇w are accurately computed with 2-point formulas. Note also that due to the staggering of the discrete velocities, the discretization of $\partial w/\partial x$ and $\partial w/\partial y$ involves co-planar unknowns (on planes of constant index k), such that only the 2D DCM formula of Sect. 2 is needed.

In addition, the DCM discretization enables us to revisit the discretization of the viscous forces at the immersed boundary by providing accurate formula for the rate-of-strain components which could not be discretized by standard 2-point formulas [2, 4]. Since $\boldsymbol{n}^{\mathrm{ib}} \cdot \boldsymbol{e}_z = 0$ in 3D geometries extruded in the z direction, the streamwise (drag) and spanwise (lift) forces at the boundary read respectively :

$$F_x = \mu \int_{\Gamma^{\mathrm{ib}}} \dot{\gamma}_{xx}\, \boldsymbol{e}_x \cdot \boldsymbol{n}\, \mathrm{d}S + \mu \int_{\Gamma^{\mathrm{ib}}} \dot{\gamma}_{xy}\, \boldsymbol{e}_y \cdot \boldsymbol{n}\, \mathrm{d}S, \tag{27a}$$

$$F_y = \mu \int_{\Gamma^{\mathrm{ib}}} \dot{\gamma}_{xy}\, \boldsymbol{e}_x \cdot \boldsymbol{n}\, \mathrm{d}S + \mu \int_{\Gamma^{\mathrm{ib}}} \dot{\gamma}_{yy}\, \boldsymbol{e}_y \cdot \boldsymbol{n}\, \mathrm{d}S, \tag{27b}$$

where μ is the dynamic viscosity and the components of the rate-of-strain tensor are $\dot{\gamma}_{xx} = 2\partial u/\partial x$, $\dot{\gamma}_{xy} = \partial u/\partial y + \partial v/\partial x$ and $\dot{\gamma}_{yy} = 2\partial v/\partial y$. For all type of cut-cells, the discretization of the drag and lift components reads :

$$F_x \cong \mu \, [\Delta S_x]^{\mathrm{ib}}_{i,j,k} \; \dot{\gamma}_{xx}\big|^{\mathrm{ib}}_{i,j,k} + \mu \, [\Delta S_y]^{\mathrm{ib}}_{i,j,k} \; \dot{\gamma}_{xy}\big|^{\mathrm{ib}}_{i,j,k}\,, \tag{28a}$$

$$F_y \cong \mu \, [\Delta S_x]^{\mathrm{ib}}_{i,j,k} \; \dot{\gamma}_{xy}\big|^{\mathrm{ib}}_{i,j,k} + \mu \, [\Delta S_y]^{\mathrm{ib}}_{i,j,k} \; \dot{\gamma}_{yy}\big|^{\mathrm{ib}}_{i,j,k}\,, \tag{28b}$$

with face-area vector components $[\Delta S_x]^{\mathrm{ib}}_{i,j,k} = (\theta^u_{i-1,j} - \theta^u_{i,j})\Delta y_j \Delta z_k$ and $[\Delta S_y]^{\mathrm{ib}}_{i,j,k} = (\theta^v_{i,j-1} - \theta^v_{i,j})\Delta x_i \Delta z_k$, and the discretization of the normal rate-of-strains $\dot{\gamma}_{xx}\big|^{\mathrm{ib}}_{i,j,k}$ $\dot{\gamma}_{yy}\big|^{\mathrm{ib}}_{i,j,k}$ are given by formulae based on the Green-Gauss theorem, valid for all type of cut-cells [1, 4]. In contrast, the discretization of the shear component $\dot{\gamma}_{xy}\big|^{\mathrm{ib}}_{i,j,k}$ has to be adapted to each type of cut-cells by means of the DCM discretization. Following the notations of Fig. 10, the discretization on the solid face of pentagonal cut-cell $\Omega_{i,j,k}$ is given by the symmetric formula :

$$\dot{\gamma}_{xy}\big|^{\mathrm{ib}}_{i,j,k} \cong \frac{1}{2}\left[\frac{\partial u}{\partial y}\bigg|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{NN}}}_{i-1,j,k} + \frac{\partial v}{\partial x}\bigg|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{NN}}}_{i,j,k} \right] + \frac{1}{2}\left[\frac{\partial u}{\partial y}\bigg|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{N}}}_{i,j,k} + \frac{\partial v}{\partial x}\bigg|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{N}}}_{i,j-1,k} \right], \tag{29}$$

where the DCM formulas at point $\boldsymbol{x}_{\mathrm{N}}$ involve discrete velocities $u_{i,j,k}$ and $v_{i,j-1,k}$ using formulas in the Cartesian basis analogous to (11), for example :

$$\frac{\partial u}{\partial y}\bigg|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{N}}}_{i,j,k} \cong \sin\beta^{\mathrm{ib}}_{i,j,k} \, \frac{u_{\mathrm{N}} - u_{i,j,k}}{\cos\alpha^{\mathrm{ib}}_{i,j,k} d^{\mathrm{ib}}_{i,j,k}} + \left(\cos\beta^{\mathrm{ib}}_{i,j,k} - \sin\beta^{\mathrm{ib}}_{i,j,k} \tan\alpha^{\mathrm{ib}}_{i,j,k}\right) \frac{u_{\mathrm{NN}} - u_{\mathrm{N}}}{\|\boldsymbol{x}_{\mathrm{NN}} - \boldsymbol{x}_{\mathrm{N}}\|}. \tag{30}$$

In a similar fashion, the DCM formulas at point $\boldsymbol{x}_{\mathrm{NN}}$ involve discrete velocities $v_{i,j,k}$ and $u_{i-1,j,k}$.

It is interesting to observe that for the computation of $\partial u/\partial y|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{N}}}_{i,j,k}$ at point $\boldsymbol{x}_{\mathrm{N}}$, since the solid boundary of $\Omega_{i,j,k}$ cut the eastern face where $u_{i,j,k}$ is positioned, the DCM direction $\boldsymbol{d}^{\mathrm{ib}}_{i,j,k}$ is parallel to $\boldsymbol{e}_y$. In this case, the solid boundary inclination and the angle of non-orthogonality verify $\beta^{\mathrm{ib}}_{i,j,k} + \alpha^{\mathrm{ib}}_{i,j,k} \equiv \mathrm{mes}(\boldsymbol{e}_x, \boldsymbol{n}^{\mathrm{ib}}_{i,j,k}) + \mathrm{mes}(\boldsymbol{n}^{\mathrm{ib}}_{i,j,k}, \boldsymbol{d}^{\mathrm{ib}}_{i,j,k}) = \pi/2$ (see Fig. 10), which allows Eq. (30) to simplify to :

$$\left.\frac{\partial u}{\partial y}\right|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{N}}}_{\mathrm{i,j,k}} \cong \frac{u_{\mathrm{N}} - u_{i,j,k}}{d^{\mathrm{ib}}_{i,j,k}}, \qquad d^{\mathrm{ib}}_{i,j,k} = \theta^{u}_{i,j,k}\Delta y_j, \tag{31}$$

showing that in this case the DCM discretization is independent of the solid boundary inclination. An analogous simplification can be performed for $\partial v/\partial x|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{NN}}}_{i,j,k}$, but not for $\partial u/\partial y|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{NN}}}_{i-1,j,k}$ or $\partial v/\partial x|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{N}}}_{i,j-1,k}$ where the DCM direction is not parallel to $\boldsymbol{e}_x$ or $\boldsymbol{e}_y$. As a matter of fact, Formula (31) is identical to the Ghost Fluid Method (GFM) used in [22] for discretizing the Poisson equation in irregular geometries. In previous works on the LS-STAG method [1, 2, 4], the GFM was employed to calculate $\partial u/\partial y|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{N}}}_{i,j,k}$ and $\partial v/\partial x|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{NN}}}_{i,j,k}$, while *ad hoc* approximations were used for $\partial u/\partial y|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{NN}}}_{i-1,j,k}$ and $\partial v/\partial x|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{N}}}_{i,j-1,k}$ (for example, Ref. [4] used a discrete version of the Schwartz identity). In this respect, the Diamond Cell Method can be viewed as an extension of the Ghost Fluid Method for computing accurately gradients at irregular boundaries.

For completeness, we give the discretization of the shear component of the rate-of-strain for the 2 other types of cut-cells. For trapezoidal cut-cell $\Omega_{i+1,j,k}$ in Fig. 10, the shear rate of strain is :

$$\dot{\gamma}_{xy}|^{\mathrm{ib}}_{i+1,j,k} \cong \frac{1}{2}\left[\left.\frac{\partial u}{\partial y}\right|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{N}}}_{i,j,k} + \left.\frac{\partial v}{\partial x}\right|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{N}}}_{i+1,j-1,k}\right] + \frac{1}{2}\left[\left.\frac{\partial u}{\partial y}\right|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{NE}}}_{i+1,j,k} + \left.\frac{\partial v}{\partial x}\right|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{NE}}}_{i+1,j-1,k}\right], \tag{32}$$

In the case where $\Omega_{i+1,j,k}$ is a triangular cut-cell such its right edge cuts the south face of the cell, only velocities $u_{i,j,k}$ and $v_{i+1,j-1,k}$ are present in the cut-cell, and the shear rate-of-strain is given by the symmetric formula :

$$\dot{\gamma}_{xy}|^{\mathrm{ib}}_{i+1,j,k} \cong \left.\frac{\partial u}{\partial y}\right|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{N}}}_{i,j,k} + \left.\frac{\partial v}{\partial x}\right|^{\mathrm{ib},\boldsymbol{x}_{\mathrm{NE}}}_{i+1,j-1,k}. \tag{33}$$

Finally, we mention that the discrete shear rates (29), (32) and (33) involve DCM gradients evaluated at distinct endpoints of the solid boundaries, in order to highlight the underlying symmetry of the formulas. As mentioned in Sect. 3.1, DCM gradients are independent of the location on the boundary where they are evaluated. Thus, for ease of implementation in the LS-STAG code, the shear rate is evaluated at the center of the solid part of the cut-cells.

The DCM discretization is tested on the benchmark flow 3D-3Z proposed by [18], which concerns unsteady laminar flow past a circular cylinder in a confined square duct. The flow unsteadiness is induced by a time-varying inflow boundary condition, yielding a time-periodic Reynolds number $\mathrm{Re} \in [0, 100]$. This flow has previously been computed in [4] with the LS-STAG/2PT method, where its performances were compared against an unstructured commercial solver (ANSYS FLUENT©) with comparable discretization schemes and computational meshes. Although LS-STAG/2PT compared favorably in terms of wall time and memory usage, inferior results were observed on the computation of the drag and lift forces. We refer to [4] for a complete description of the geometry, boundary conditions, meshes and discretizations used by both solvers.

Fig. 11 compares the drag and lift computations on one-half period of the simulation ($t \in [0, 8\mathrm{s}]$) obtained by the LS-STAG/2PT and LS-STAG/DCM methods on a series of refined meshes M1-M3 (500,000 to 3M cells). LS-STAG/DCM clearly displays a superior accuracy on each mesh level, most notably after $t \simeq 3\,\mathrm{s}$ where the

	% L_2 Error on C_{D}			% L_2 Error on C_{L}			Wall time (min)		
	M_2	M_3	M_4	M_2	M_3	M_4	M_2	M_3	M_4
LS-STAG/2PT [4]	3.74	2.42	3.05	48.06	16.94	10.22	15	125	2013
LS-STAG/DCM	1.48	0.65	0.22	41.91	10.06	5.46	17	138	2241
FLUENT© [4]	2.11	0.81	0.21	17.03	10.01	2.67	55	488	7841

Table 2: Performances of the LS-STAG and FLUENT© computations in terms of computational accuracy (normalized errors on the force coefficients with respect to the reference solution [23] for one flow period) and wall time (the computations have been performed on the IJL-LEMTA parallel machine with 2 processors for the M2 mesh and 4 processors for the M3 and M4 meshes).

force signals undergo rapid variations. A more quantitative comparison is performed in Table 2, that compares the computational accuracy (L^2 errors on the force coefficients) and the wall time of the 2 variants of LS-STAG and the FLUENT© computations [4]. As observed in this reference the LS-STAG/2PT solver gives only monotonic convergence for the lift quantity, while the drag coefficient fails to convergence. In contrast, the DCM discretization show a dramatic increase in the accuracy of the LS-STAG discretization, yielding a smooth convergence for all force coefficients, with an accuracy comparable to the FLUENT© solver on all 3 grids. In terms on run-time efficiency, LS-STAG/DCM is only 10% more CPU intensive that LS-STAG/2PT, due to the need of vertex reconstruction and the use of a wider stencil for the linear solves at each time-step. Both variants of LS-STAG are about 300% faster than FLUENT©. These results show that a cut-cell method with DCM gradients can be a promising kernel, in terms of computational cost and accuracy, to traditional body-fitted solvers for flows in complex geometries.

5. Conclusion and future work

This paper has presented the application of the diamond cell method (DCM) for improving the accuracy of diffusion in the LS-STAG cut-cell method compared to the standard 2-point approximation. The DCM has been employed for cell-centered data (temperature equation in 2D) and face-centered data (Navier-Stokes equations in 3D-extruded geometries), which illustrates the versatility of the approach, that can be applied to other cut-cell methods. Simple and effective strategies for implementing Dirichlet and Neumann boundary conditions are detailed, and have shown no impact on the global accuracy of the method. One of the main findings of the paper is that the accuracy of the DCM approximation is highly dependent on the interpolation scheme that is used, otherwise inaccurate results are encountered in the smaller cut-cells. In particular, the barycentric interpolation that we propose is able to obtain local accuracy in cut-cells comparable to the one reached in Cartesian cells, irrespective of the size and shape of the cut-cell. Finally, the benchmark flow in extruded geometry with DCM discretization has shown a dramatic increase in accuracy compared to the 2-point discretization, for comparable computational costs.

Progress is on the way to employ DCM for full 3D computations with the LS-STAG method. The DCM framework will enable us to formulate a systematic discretization in the cut-cells that alleviates the need to adapt the discretization to each different type of cut-cells, as it was done in [4] for 3D-extruded geometries. For all components of the velocity gradients, only the 2D version of DCM is needed, since velocities are stored at face centers. However, for scalar quantities such as temperature which are stored at cell centers, a 3D extension of the DCM [24] needs to be employed.

References

[1] Y. Cheny, O. Botella, The LS-STAG method : A new immersed boundary / level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties, J. Comput. Phys. **229**, 1043-1076.

[2] O. Botella, M. Ait-Messaoud, A. Pertat, C. Rigal, Y. Cheny, The LS-STAG immersed boundary method for non-Newtonian flows in irregular geometries : Flow of shear-thinning liquids between eccentric rotating cylinders., Theoretical and Computational Fluid Dynamics **29**, 93-110.

[3] O. Botella, Y. Cheny, F. Nikfarjam, M. Stoica, Application of the LS-STAG immersed boundary/cut-cell method to viscoelastic flow computations, Communications in Computational Physics 20 (2016) 870–901.

[4] F. Nikfarjam, Y. Cheny, O. Botella, The LS-STAG immersed boundary/cut-cell method for non-Newtonian flows in 3D extruded geometries, Computer Physics Communications 226 (2018) 67–80.

[5] L. Martinelli, A. Jameson, Validation of a multigrid method for the Reynolds averaged equations, AIAA paper No. 88-0414 (1988).

[6] F. Jacon, D. Knight, A Navier-Stokes algorithm for turbulent flows using an unstructured grid and flux difference splitting, AIAA paper No. 94-2292 (1994).

[7] J. H. Ferziger, M. Periĉ, Computational Methods for Fluid Dynamics, Springer, Berlin, 1996.

[8] S. Mathur, J. Murthy, A pressure-based method for unstructured meshes, Numerical Heat Transfer 31 (1997) 195–215.

[9] F. Moukalled, L. Mangani, M. Darwish, The Finite Volume Method in Computational Fluid Dynamics, Springer, 2016.

[10] Y. Coudière, J.-P. Vila, P. Villedieu, Convergence rate of a finite volume scheme for a two dimensional convection-diffusion problem, ESAIM: Mathematical Modelling and Numerical Analysis 33 (1999) 493–516.

[11] E. Bertolazzi, G. Manzini, A cell-centered second-order accurate finite volume method for convection–diffusion problems on unstructured meshes, Mathematical Models and Methods in Applied Sciences 14 (2004) 1235–1260.

[12] E. Bertolazzi, G. Manzini, On vertex reconstructions for cell-centered finite volume approximations of 2D anisotropic diffusion problems, Mathematical Models and Methods in Applied Sciences 17 (2007) 1–32.

[13] R. Falgout, A. Cleary, J. Jones, E. Chow, V. Henson, C. Baldwin, P. Brown, P. Vassilevski, U. M. Yang, Center for Applied Scientific Computing (CASC), Lawrence Livermore National Laboratory. [link].
URL `http://acts.nersc.gov/hypre/`

[14] I. Demirdžić, Ž. Lilek, M. Perić, Fluid flow and heat transfer test problems for non-orthogonal grids: Bench-mark solutions, International Journal for Numerical Methods in Fluids 15 (3) (1992) 329–354.

[15] F. Sotiropoulos, X. Yang, Immersed boundary methods for simulating fluid–structure interaction, Progress in Aerospace Sciences 65 (2014) 1–21.

[16] T. Ye, R. Mittal, H. S. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys. **156**, 209-240.

[17] M. P. Kirkpatrick, S. W. Armfield, J. H. Kent, A representation of curved boundaries for the solution of the

Navier-Stokes equations on a staggered three-dimensional Cartesian grid, Journal of Computational Physics **184**, 1-36.

[18] E. Bayraktar, O. Mierka, S. Turek, Benchmark computations of 3D laminar flow around a cylinder with CFX, OpenFOAM and FeatFlow, International Journal of Computational Science and Engineering 7 (3) (2012) 253–266.

[19] P. van der Plas, H. van der Heiden, A. Veldman, R. Luppes, R. Verstappen, Efficiently simulating viscous flow effects by means of regularization turbulence modeling and local grid refinement, Seventh International Conference on Computational Fluid Dynamics (ICCFD7), Hawaii, paper ICCFD-2503, 2012.

[20] L. Martinelli, Calculations of viscous flows with a multigrid method, Ph.D. thesis, Princeton University (1987).

[21] I. Amidror, Scattered data interpolation methods for electronic imaging systems: A survey, Journal of electronic imaging 11 (2002) 157–176.

[22] F. Gibou, R. P. Fedkiw, L.-T. Cheng, M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, J. Comput. Phys. **176**, 205-227.

[23] E. Bayraktar, Private communication (2016).

[24] Y. Coudière, G. Manzini, Benchmark 3D: The cell-centered finite volume method using least squares vertex reconstruction ("diamond scheme"), in: Finite Volumes for Complex Applications VI Problems & Perspectives, 2011, pp. 985–992.

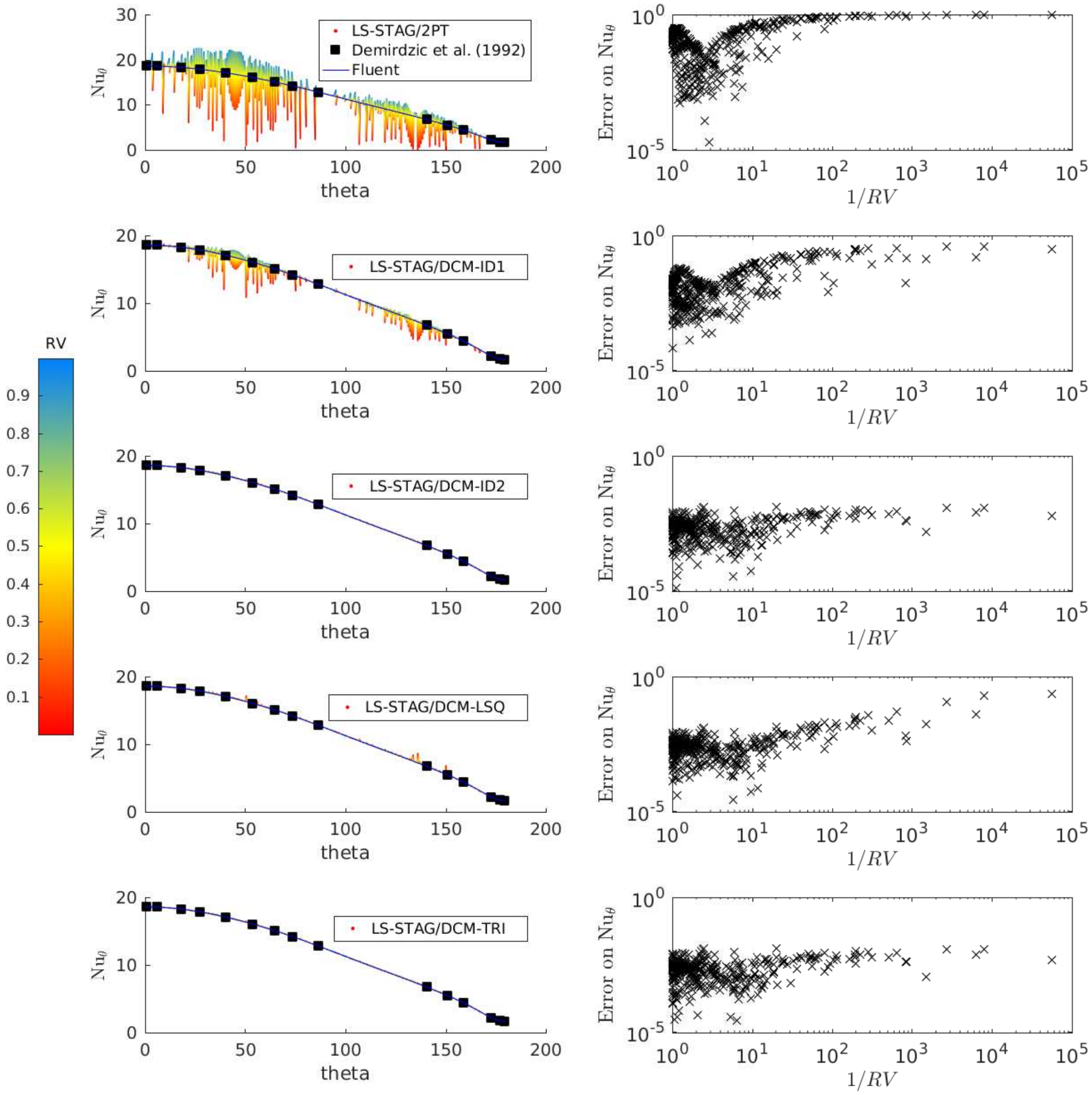

Figure 9: At left, from top to bottom: Profiles of the Nusselt number along the half cylinder for the various discretization of diffusion in the LS-STAG method, and comparison with the FLUENT$^{\copyright}$ computation and reference results of Demirdžić *et al* [14]. The LS-STAG profiles are colorized with respect to the volume-change ratio $\mathrm{RV}_{i,j}$ (25) of the cut-cells. At right: corresponding cell-wise local errors as a function of $\mathrm{RV}_{i,j}$ (the reference is the FLUENT L5 solution).

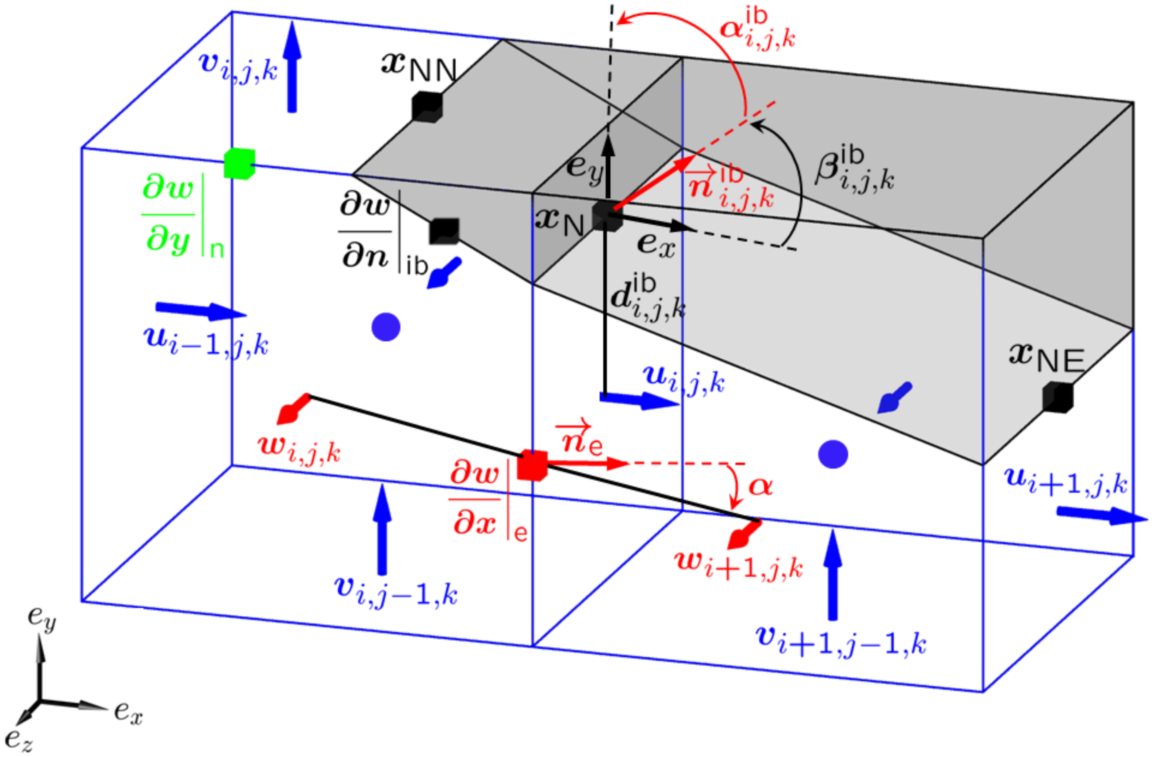

Figure 10: Staggered mesh for 3D extruded geometries in the z direction, and location of the off-diagonal components of ∇w.

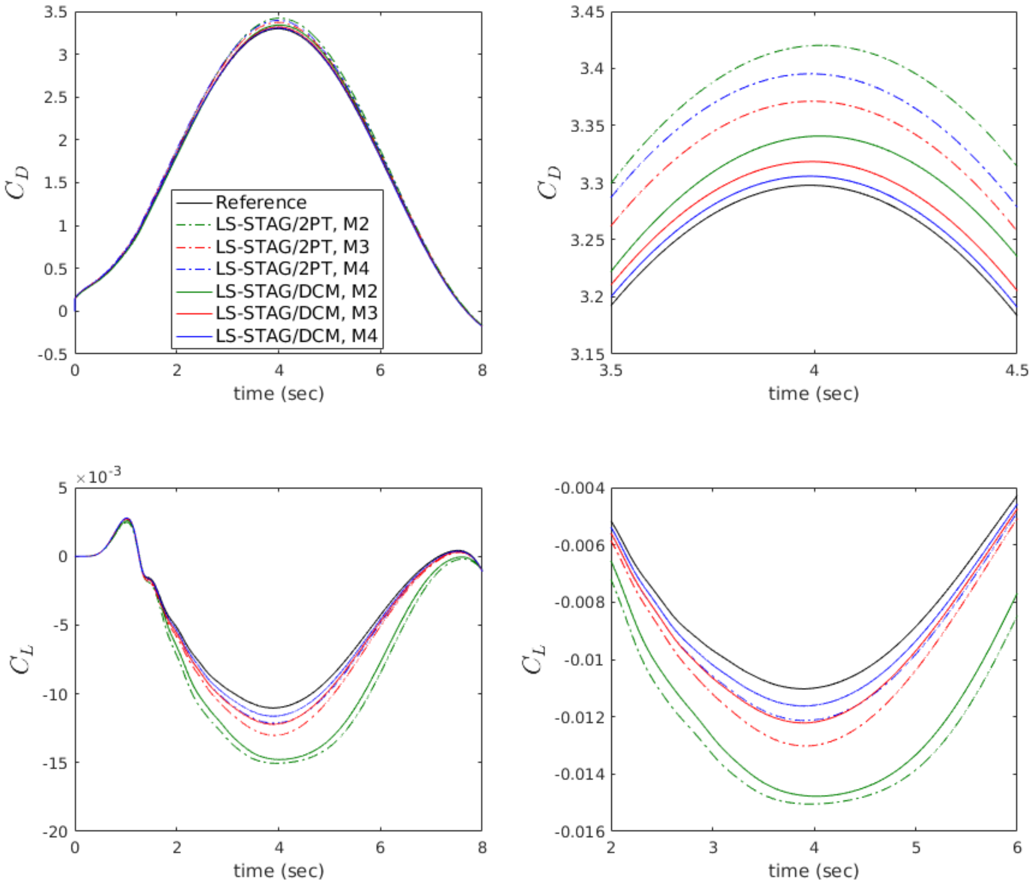

Figure 11: Time evolution of the force coefficients computed by the 2 variants of the gradient computations and reference solution [23]. At left: on the total time interval; at right: close up around $t = 4\,\mathrm{s}$.